

---

**COMPUTER SUBJECT:** REST PROVIDER AND CONSUMER

**TYPE:** GROUP WORK EXERCISE/DISCUSSION

**IDENTIFICATION:** Another REST / Pele

**COPYRIGHT:** *Peter Levinsky*

**LEVEL:** MIDDLE

**DURATION:** 3-4 hours

**SIZE:** 100-140 lines!! (Seven assignments)

**OBJECTIVE:** Stabilize and rooted REST Provider and REST consumer

**REQUIREMENTS:** REST

**COMMANDS:**

---

IDENTIFICATION: Another REST /Pele

## Programming C#

---

### The Mission

We have an existing database containing information on apartment selling in Copenhagen. The apartments have information of location, price, no of rooms, size and if there is a washing machine or dishwasher.

The idea is to provide a REST full service of this information and to create a small program to consume some of the services.

### Assignment 1: Create a table in your database I Azure:

In Azure you have a database. Create a table '**Apartment**' having the columns

- Id (int)
- Price(int)
- Location(nvarchar(30))
- PostalCode(int)
- Size(int)
- NoRoom(int)
- WashingMachine(bit)
- Dishwasher(bit)

Add some rows to your table.

### Assignment 2: First step in REST Service:

Create a new project ApartmentRest (a WCF-project).

Rename the IService1 to IApartmentService and the Service1 to ApartmentService (If you can),

Make some simple reading (GET) request:

- Get all apartments (i.e. './apartments' ) with a method  
`public IList<Apartment> GetAllApartment()`
- Get all apartments for a specific PostalArea (i.e. './apartments/postal/{code}' )  
`public IList<Apartment> GetAllApartmentByPostalCode(string code)`
- Get all apartments for a specific location or contain part of it  
(i.e. './apartments/location/{location}' )  
`public IList<Apartment> GetAllApartmentByLocation(string location)`

Design and code the Interface (IApartmentService) to support this three accessing of resources from the database. Remember to define the annotations '[ServiceContract] [OperationContract] [WebInvoke]', for WebInvoke set the Method, RequestFormat (if needed), ResponseFormat (if needed), UriTemplate.

(Hint: You can use Entity Framework to access the table in the database – to generate an Apartment – class – though this has the disadvantage it will be more difficult to test in UnitTest )

---

### Assignment 3: Create a REST Consumer

Create a new project ApartmentRestConsumer (a console-project).

Demonstrate the above three REST services in the main-method. You do not need to do any fancy UI just plain Console.Writeline will go.

(Hint: You need to define and implement an apartment-class)

### Assignment 4: Modify : REST service

Modify your IApartmentService and ApartmentService to support create, delete and update.

- Create an Apartment (POST) (i.e. . ‘.../apartments’)  
public void CreateApartment(Apartment apartment)
- Delete an Apartment (DELETE) (i.e. . ‘.../apartments/{id}’)  
public void DeleteApartment(string id) // delete the apartment with specified id
- Update an Apartment (UPDATE) (i.e. . ‘.../apartments/{id}’)  
public void UpdateApartment(string id, Apartment apartment) // update the apartment with the specified id

### Assignment 5: Check REST servive

Use Fiddler or Postman to check the 6 services (3\*Get, Post, Put and Delete)

### Assignment 6: Modify your Console application (Consumer)

Modify your Console application to support create, delete and update apartments.

### Assignment 7: Test your REST-full service

Create a UNIT-test of your ApartmentService, and test your application to validate it is working proper.

(Hint if you have used the entity framework:

- 1) Copy connectionstring from web.config to app.config(if not in testproject then add->new item->Application configuration)
  - 2) For references then add reference EntityFramework.dll + EntityFramework.SqlServer.dll
- )

---

*Assignment 8: Modify : REST service to support searching*

Modify your IApartmentService and ApartmentService to support search.

You should be able to search for apartments in a price range (minPrice, maxPrice).

You should be able to search for apartment, which have washingMachine and / or dishwasher.

Try your new services from a browser.